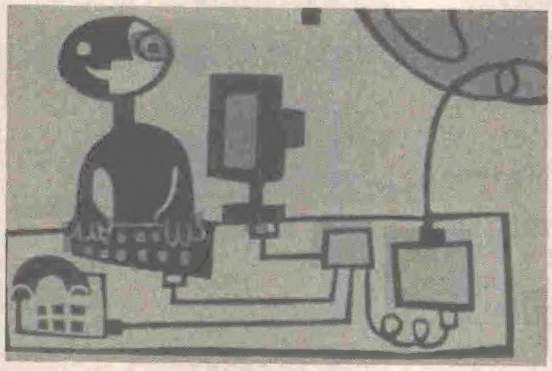
ประพนธ์ อัศวภาณุวัฒน์

SOFT MAIL

OLE Technology ตอนที่ 1

สิ่งหนึ่งที่ช่วยทำให้ระบบปฏิบัติวินโดวส์ประสบความสำเร็จคือเทคโนโลยี OLE



สวัสดีครับ ในปัจจุบันอาจกล่าวได้ว่า ระบบปฏิบัติการวินโดวส์เป็นระบบที่มีการใช้งานแพร่หลายมากที่สุด ปัจจัยที่ทำให้ระบบปฏิบัติการนี้มีความนิยมใช้งานอย่างแพร่หลาย นอกเหนือจากการใช้งานที่ง่ายแล้ว คือ ความทรงประสิทธิภาพของระบบการทำงานเอง และสิ่งหนึ่งที่ช่วยทำให้ระบบปฏิบัติการตัวนี้ประสบความสำเร็จคือ OLE (Object Linking and Embedding) Technology ซึ่งจะเป็นหัวข้อที่ผมขอกล่าวถึงในฉบับนี้

OLE คืออะไร เป็นคำถามที่ตอบค่อนข้างยากครับ ถ้ายึดจุดประสงค์ดั้งเดิมของเทคโนโลยีตัวนี้ (เวอร์ชัน 1.0) เราสามารถจำกัดความได้ว่าเป็น Object Linking และ Object Embedding ซึ่งเป็นเทคโนโลยีที่อยู่บนพื้นฐานของ COM (Component Object Module) เหตุที่ผมต้องกล่าวเช่นนี้เนื่องจากเทคโนโลยีตัวนี้มีวิวัฒนาการแบบก้าวหน้า กล่าวคือ ณ ปัจจุบันนี้ ถ้าพูดถึงเฉพาะเรื่อง Linking และ Embedding แล้ว สิ่งนี้จะเป็นเพียงองค์ประกอบส่วนหนึ่งของ OLE เวอร์ชันปัจจุบันเท่านั้น

สรุปแล้ว OLE ไม่ได้เป็นคำย่อของ Object Linking และ Object Embedding แต่เป็นคำนิยามชื่อเรียกเทคโนโลยีด้านการสื่อสารและเชื่อมโยงของออบเจ็กต์

COM Object

เมื่อกล่าวถึง OLE และ ActiveX แล้ว คงเป็นการยากที่จะไม่กล่าวถึง COM Technology เอาเป็นว่าผมขอกล่าวสรุปโดยสั้น และจะขอเจาะลึกอีกครั้ง เมื่อมีโอกาสในบทความชุดต่อไปครับ

COM คือออบเจ็กต์ชนิดที่เรียกได้ว่าเป็น client/server object-base module ที่ออกแบบโดยไมโครซอฟท์ โดยใช้สนับสนุน OLE, ActiveX, DirectX เทคโนโลยี โดยเป็นแนวความคิดร่วมของส่วนระบบปฏิบัติการใน WinTel Platform (Window/Intel) เพื่อที่จะสร้างชุด Binary Objects ที่มีความสามารถในส่วนของ Language-independent framework, Remote Accessible และ Platform-independent เป็นหลัก

Language-independent framework เป็นเทคโนโลยีที่ไม่ยึดติดกับตัวภาษา เราสามารถใช้ภาษาใด ๆ ที่สนับสนุน COM Object สร้างออบเจ็กต์ชนิดนี้ขึ้นมา โดยปฏิบัติตามกฎและข้อตกลงที่มีไว้สำหรับการกำหนดกิจกรรมของ COM Object และทำนองเดียวกัน COM Object ที่สร้างขึ้นจะสามารถนำไปใช้กับตัวแปรภาษาอื่น ๆ ที่สนับสนุน COM Object ได้ทันที

Remote Accessible ความสามารถในลักษณะ Remote accessible เป็นการกำหนดให้ COM Object ที่สร้างขึ้นสามารถเรียกใช้โดยเครื่องคอมพิวเตอร์เครื่องอื่นโดยผ่านระบบ Network Environment ชนิดต่าง ๆ เราเรียกออบเจ็กต์ชนิดนี้ว่า Distributed-COM (หรือ DCOM) ในปัจจุบันระบบปฏิบัติการที่สนับสนุน DCOM คือ Windows NT

Platform-independent ถือว่าเป็นความฝันอันสูงสุดของไมโครซอฟท์ ในการที่จะนำ COM Object ไปใช้งานในต่าง ๆ สภาวะแวดล้อม นอกเหนือจากระบบปฏิบัติการตระกูลวินโดวส์ของตนเอง

ถ้าจะให้เปรียบเทียบแล้ว COM Object ที่เป็นพื้นฐานของ OLE และ ActiveX ก็เปรียบเสมือนเป็น Tobject Class ของ Delphi ที่เป็น base class ให้ Objects อื่น ๆ สืบสกุลเพื่อสร้างเป็น Components หรือ Objects ใหม่ แต่ต่างกันตรงที่ COM Object จะไม่มี Propertys และ Events ให้เรียกใช้ และการติดต่อสื่อสารระหว่างกันของ COM Object จะต้องสื่อสารกันผ่านอินเตอร์เฟสในรูปแบบ Client/Server Object ที่กำหนดไว้เท่านั้น

OLE ทำงานอย่างไร

OLE Object จะสื่อสารกับแอพพลิเคชันที่เรียกใช้ในลักษณะ Client/Server กล่าวคือ เมื่อมีการเรียกใช้ OLE Object จากแอพพลิเคชันหนึ่ง ๆ OLE Object จะทำหน้าที่เป็น Server Object และแอพพลิเคชันจะทำหน้าที่เป็น Client Object โดย OLE Object จะ active ภายใน Container ของแอพพลิเคชัน โดยจะเกิดการรวมตัวระหว่าง OLE Object และแอพพลิเคชัน เช่น รวมเมนูของแอพพลิเคชันและ OLE Object เข้าด้วยกัน หน้าต่างของแอพพลิเคชันก็จะกลายเป็นหน้าต่างของ OLE Object ทันที (ถ้าเป็น OLE 1.0 แล้ว OLE Object จะทำการสร้างหน้าต่างใหม่ โดยมีข้อมูลของแอพพลิเคชันอยู่ภายใน และพร้อมที่จะใช้งาน)

Linking และ Embedding Objects

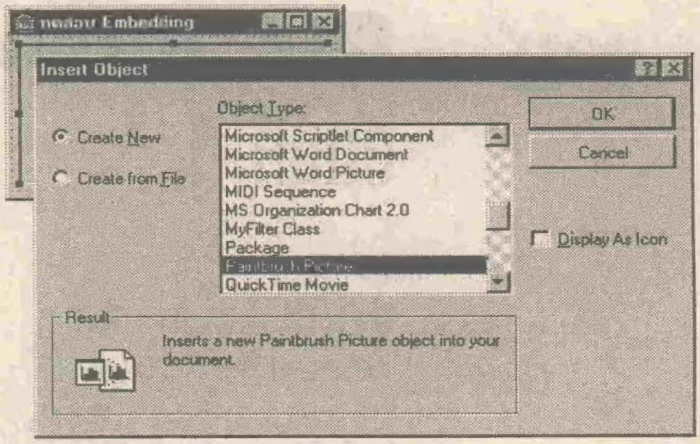
Linking จะเกิดหลังจากที่เราเปิด OLE Object File ด้วย OLE Container การเรียกใช้งานในลักษณะนี้ Object File จะไม่ถูกบรรจุใน OLE Container แต่สร้าง Path เชื่อมโยงระหว่าง Container และ Object File ประโยชน์ที่ได้รับจากการ Linking คือสามารถใช้งานในลักษณะ Multi-Link ทำให้สามารถ Link แอพพลิเคชันหลาย ๆ โปรแกรมเข้ากับ OLE Object File ตัวเดียว เมื่อข้อมูลที่อยู่ใน OLE Object File เปลี่ยน ข้อมูลที่แอพพลิเคชันทุกตัวที่ Link อยู่ก็จะเปลี่ยนตามไปด้วย

Embedding แตกต่างกับ Linking ตรงที่เมื่อมีการเรียกใช้ด้วยวิธีนี้ OLE Object จะถูกเก็บอยู่ใน OLE Container ของตัวแอพพลิเคชันเอง ทำให้แอพพลิเคชันอื่น ๆ ไม่สามารถติดต่อกับ Object File ที่ Embedding อยู่กับแอพพลิเคชันแรกได้ ทำให้เป็นการปกป้องข้อมูลที่ระดับหนึ่ง

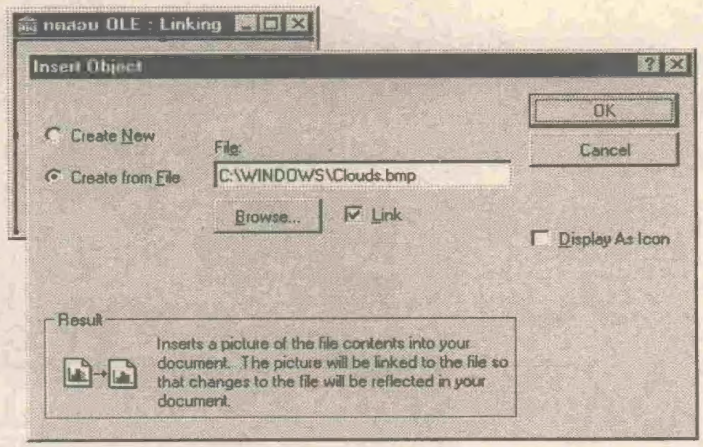
ToleContainer

Delphi ใช้ ToleContainer Component เป็น OLE Server สำหรับรองรับการใช้งานในส่วนของ OLE Container โดยจะบรรจุเป็น VCL มาตรฐานอยู่ในแถบ System Page ของ Component Pallete การใช้งาน ToleContainer สามารถทำได้ดังต่อไปนี้

ขั้นตอนการ Embedding OLE Object ด้วยหน้าต่าง Insert Object:



รูปที่ 1 การใช้ OLE Embedding

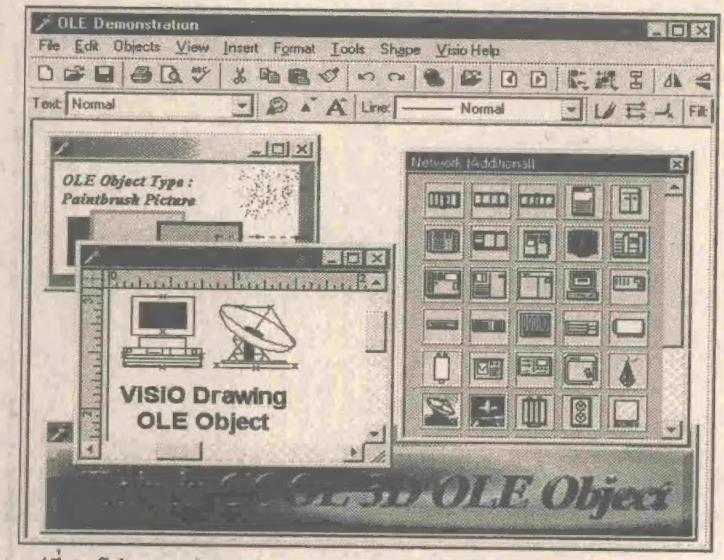


รูปที่ 2 การใช้ OLE Linking

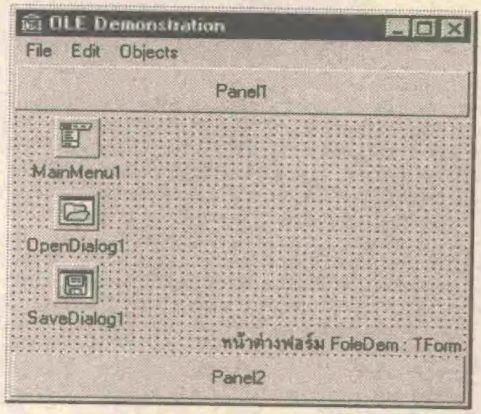
1. เพิ่ม ToleContainer Component ลงในฟอร์ม จัดหน้าจอโดยกำหนดให้ OleContainer1 มี align เป็น alClient
2. ดับเบิลคลิกที่พื้นที่ของ ToleContainer Objects เพื่อรันโปรแกรม Insert Object Window ดังรูปที่ 1
3. เลือก Paintbrush Picture แล้วกดปุ่ม [OK]
4. ดับเบิลคลิกที่กรอบ OleContainer1 เพื่อทำการแก้ไขรูปภาพโดยตรง หรือสั่งรันโปรแกรม (F9) และดับเบิลคลิกที่กรอบรูป

ขั้นตอนการ Linking OLE Object:

1. เปิดฟอร์มใหม่ เพิ่ม ToleContainer ลงในฟอร์ม
2. ดับเบิลคลิกที่ OleContainer1 เพื่อรันโปรแกรม Insert Object ดังรูปที่ 2
3. เลือกหัวข้อ "Create From File" และกาเครื่องหมาย ☑ หน้า Link
4. เลือกไฟล์ที่ต้องการโดยกดปุ่ม "Browse.."
5. รันโปรแกรมโดยดับเบิลคลิกที่กรอบรูป



รูปที่ 3 โปรแกรม OLEDEM.DPR แสดงการ Embedding กับเอกสารชนิดต่าง ๆ ในขณะเดียวกัน



รูปที่ 4 หน้าต่างฟอร์ม FoleDem

โปรแกรมตัวอย่าง : OLEDEM

OLEDEM.DPR ดังรูปที่ 3 เป็นโปรแกรมตัวอย่างที่ช่วยเสริมความเข้าใจในหลักการทำงานของ OLE Object

สำหรับโปรแกรมนี้ผมออกแบบโดยใช้หน้าต่างแบบ MDI มีจุดประสงค์เพื่อที่สามารถเปิดหน้า OLE สำหรับเอกสารต่าง ๆ ได้พร้อม ๆ กัน โดยกำหนดให้หน้าต่าง MDI Child เป็นส่วนที่บรรจุ OLE Container และหน้าต่าง MDI Form ซึ่งเป็นหน้าต่างหลักที่จะมีการเปลี่ยนแปลงระบบเมนู และ Tool Bar ตามชนิดของ OLE Object ที่ติดต่อด้วย มีส่วนของโปรแกรมดังต่อไปนี้

โปรแกรมหลัก : uOLEDem.PAS

uOLEDem.pas ดังรูปที่ 4 แสดงอยู่ในลิสติงที่ 1 เป็นส่วนของการควบคุม OLE ทั้งหมด มีกิจกรรมและส่วนของโปรแกรมที่สำคัญดังต่อไปนี้

ลิสติงที่ 1 โปรแกรม uOLEDEM.PAS

```
00001:  // *************************
00002:  // uOLEDEM.PAS
00003:  //   Release date : 11/11/41
00004:  // *************************
00005:  unit uOLEDEM;
00006:
00007:  interface
00008:
00009:  uses
00010:       Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00011:    Menus, ExtCtrls, OleCtnrs, uOLEWnd;
00012:
00013:  type
00014:    TFoleDem = class(TForm)
00015:      Panel1: TPanel;
00016:      MainMenu1: TMainMenu;
00017:      Panel3: TPanel;
00018:      File1: TMenuItem;
00019:      Edit1: TMenuItem;
00020:      Objects1: TMenuItem;
00021:      Install1: TMenuItem;
00022:      Load1: TMenuItem;
00023:      Saveas1: TMenuItem;
00024:      Copy1: TMenuItem;
00025:      Paste1: TMenuItem;
00026:      PasteSpecial1: TMenuItem;
00027:      ChangeIcon1: TMenuItem;
00028:      DisplayProperties1: TMenuItem;
00029:      Removeobject1: TMenuItem;
```

→

ลิสติงที่ 1 (ต่อ)

```
00030:     Verbs1: TMenuItem;
00031:     Do1: TMenuItem;
00032:     Show1: TMenuItem;
00033:     Hide1: TMenuItem;
00034:     Exitprogram1: TMenuItem;
00035:     OpenDialog1: TOpenDialog;
00036:     SaveDialog1: TSaveDialog;
00037:     procedure FormCreate(Sender: TObject);
00038:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00039:     procedure Install1Click(Sender: TObject);
00040:     procedure Load1Click(Sender: TObject);
00041:     procedure Saveas1Click(Sender: TObject);
00042:     procedure Exitprogram1Click(Sender: TObject);
00043:     procedure Copy1Click(Sender: TObject);
00044:     procedure Paste1Click(Sender: TObject);
00045:     procedure PasteSpecial1Click(Sender: TObject);
00046:     procedure ChangeIcon1Click(Sender: TObject);
00047:     procedure DisplayProperties1Click(Sender: TObject);
00048:     procedure Removeobject1Click(Sender: TObject);
00049:     procedure Show1Click(Sender: TObject);
00050:     procedure Hide1Click(Sender: TObject);
00051:   private
00052:     { Private declarations }
00053:       oleObjCnt : integer;
00054:     procedure InitVar;
00055:     procedure ReleaseVar;
00056:     function  GetActiveMDIchild : TfOleWnd;
00057:   public
00058:     { Public declarations }
00059:   end;
00060:
00061: var
00062:   FoleDem: TFoleDem;
00063:
00064: implementation
00065:
00066: {$R *.DFM}
00067:
00068: //---------------------------------------------------
00069:
00070: procedure TFoleDem.InitVar;
00071: begin
00072:   oleObjCnt:=0;
00073: end;
00074:
00075: procedure TFoleDem.Releasevar;
00076: begin
00077: //
00078: end;
00079:
00080: //---------------------------------------------------
00081:
00082: function  TFoleDem.GetActiveMDIchild : TfOleWnd;
00083: var rFrm : TForm;
00084: begin
00085:   rFrm:=FoleDem.ActiveMDIChild; result:=TfOleWnd( rFrm );
00086: end;
00087:
00088: //---------------------------------------------------
00089:
00090: procedure TFoleDem.FormCreate(Sender: TObject);
00091: begin
00092:   InitVar;
00093: end;
00094:
00095: procedure TFoleDem.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00096: begin
00097:   ReleaseVar;
00098: end;
00099:
00100: procedure TFoleDem.Install1Click(Sender: TObject);
00101: begin
```

→

ลิสติงที่ 1 (ต่อ)

```
00102:    if ( oleObjCnt=MAX_OLE_CLIENTS ) then exit;
00103:    inc( oleObjCnt );
00104:    Application.CreateForm( TfOleWnd, FoleWnd[ oleObjCnt ] );
00105:    FoleWnd[ oleObjCnt ].Show;
00106:    if ( FoleDem.ActiveMDIChild=nil ) then exit;
00107:    Objects1.Enabled:=true; Edit1.enabled:=true;
00108:    GetActiveMDIchild.caption:='Loading..';
00109:    Application.ProcessMessages;
00110:    with GetActiveMDIchild do
00111:    begin
00112:      with OleContainer1 do
00113:      begin
00114:        if ( InsertObjectDialog ) then Doverb( ovShow );
00115:      end;
00116:    end;
00117:  end;
00118:
00119:  procedure TFoleDem.Load1Click(Sender: TObject);
00120:  begin
00121:    GetActiveMDIchild.caption:='Loading..';
00122:    Application.ProcessMessages;
00123:    if ( OpenDialog1.Execute ) then
00124:    begin
00125:      GetActiveMDIchild.OleContainer1.LoadFromFile( OpenDialog1.FileName );
00126:      GetActiveMDIchild.OleContainer1.Modified:=false;
00127:    end;
00128:  end;
00129:
00130:  procedure TFoleDem.Saveas1Click(Sender: TObject);
00131:  var s : string;
00132:  begin
00133:    GetActiveMDIchild.caption:='Saving..';
00134:    Application.ProcessMessages;
00135:    if ( SaveDialog1.Execute ) then
00136:    begin
00137:      s:=SaveDialog1.FileName;
00138:      if ( SaveDialog1.FilterIndex=3 ) then
00139:        GetActiveMDIchild.OleContainer1.SaveToFile( s )
00140:      else
00141:        GetActiveMDIchild.OleContainer1.SaveAsDocument( s );
00142:      GetActiveMDIchild.OleContainer1.Modified:=false;
00143:    end;
00144:  end;
00145:
00146:  procedure TFoleDem.Exitprogram1Click(Sender: TObject);
00147:  begin
00148:    FoleDem.Close;
00149:  end;
00150:
00151:  procedure TFoleDem.Copy1Click(Sender: TObject);
00152:  begin
00153:    GetActiveMDIchild.OleContainer1.Copy;
00154:  end;
00155:
00156:  procedure TFoleDem.Paste1Click(Sender: TObject);
00157:  begin
00158:    if ( GetActiveMDIchild.OleContainer1.CanPaste ) then
00159:      GetActiveMDIchild.OleContainer1.Paste;
00160:  end;
00161:
00162:  procedure TFoleDem.PasteSpecial1Click(Sender: TObject);
00163:  begin
00164:    if ( GetActiveMDIchild.OleContainer1.CanPaste ) then
00165:      GetActiveMDIchild.OleContainer1.PasteSpecialDialog;
00166:  end;
00167:
00168:  procedure TFoleDem.ChangeIcon1Click(Sender: TObject);
00169:  begin
00170:    with GetActiveMDIchild do
00171:    begin
00172:      if ( OleContainer1.state<>osEmpty ) then
00173:        OleContainer1.ChangeIconDialog;
```

→

ลิสติงที่ 1 (ต่อ)

```
00174:    end;
00175:  end;
00176:
00177:  procedure TFoleDem.DisplayProperties1Click(Sender: TObject);
00178:  begin
00179:    with GetActiveMDIchild do
00180:    begin
00181:      OleContainer1.ObjectPropertiesDialog;
00182:    end;
00183:  end;
00184:
00185:  procedure TFoleDem.Removeobject1Click(Sender: TObject);
00186:  begin
00187:    with GetActiveMDIchild do
00188:    begin
00189:      if ( OleContainer1.Modified ) then Saveas1Click(Sender);
00190:      OleContainer1.DestroyObject;
00191:      Caption:=''; Close;
00192:    end;
00193:  end;
00194:
00195:  procedure TFoleDem.Show1Click(Sender: TObject);
00196:  begin
00197:    if ( GetActiveMDIchild.OleContainer1.State<>osEmpty ) then
00198:      GetActiveMDIchild.OleContainer1.DoVerb( ovShow );
00199:  end;
00200:
00201:  procedure TFoleDem.Hide1Click(Sender: TObject);
00202:  begin
00203:    if ( GetActiveMDIchild.OleContainer1.State<>osEmpty ) then
00204:      GetActiveMDIchild.OleContainer1.DoVerb( ovHide );
00205:  end;
00206:
00207:  end.
```

function GetActiveMDIchild : TfOleWnd; เป็นกิจกรรมสำหรับอ้างถึงหน้าต่าง MDI Child ที่ active อยู่ในปัจจุบัน ด้วยคำสั่ง ActiveMDIChild และทำการ TypeCast เป็น TfOleWnd เพื่อ return ค่าให้กับกิจกรรมที่เรียกใช้

procedure TFoleDem.Install1Click(Sender: TObject); เป็นกิจกรรมสำหรับสร้างหน้าต่าง MDI Child และเปิดหน้าต่าง Insert Object Dialog Box เพื่อให้ยูสเซอร์เลือกชนิด OLE Object ที่ต้องการ และกำหนดให้แสดงด้วยกิจกรรม Doverb(ovShow)

พารามิเตอร์ที่สามารถผ่านให้กับกิจกรรม Doverb มีดังนี้

ovPrimary สร้าง OLE Server ในหน้าต่างแอพพลิเคชันปัจจุบัน

ovOpen สร้าง OLE Server เป็นหน้าต่างแยกต่างหาก

ovShow คำสั่งให้ดำเนินการสร้าง OLE Server ในหน้าต่างปัจจุบันก่อน (ovPrimary) ถ้าไม่สำเร็จให้สร้างหน้าต่างแยกแทน (ovOpen)

ovHide ถ้า OLE Server สร้าง (activated) ภายในแอพพลิเคชันจะเป็นการสั่งให้ยกเลิก (deactivated) ไม่เช่นนั้นให้ซ่อนหน้าต่าง

ovUIActivate สร้าง OLE Server เป็นหน้าต่างใหม่ภายในแอพพลิเคชัน

ovInPlaceActivate สร้าง OLE Server โดยไม่มีส่วนของ User Interface

ovDiscardUndoState ยกเลิกข้อมูล สำหรับการ undo

procedure TFoleDem.Load1Click(Sender: TObject); กิจกรรมสำหรับ Load Object File เข้าสู่ระบบ

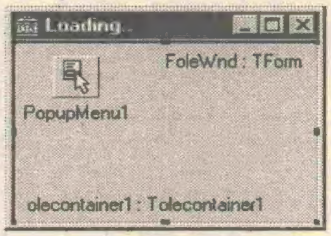
procedure TFoleDem.Saveas1Click(Sender: TObject); กิจกรรมสำหรับบันทึกข้อมูลจาก Object ที่ใช้งานอยู่ โดยถ้าเลือกชนิดของไฟล์เป็น OLE Type ให้บันทึกข้อมูลในรูปแบบ Object File

procedure TFoleDem.PasteSpecial1Click(Sender: TObject); กิจกรรมสำหรับการ paste แบบพิเศษด้วย Paste special Dialog

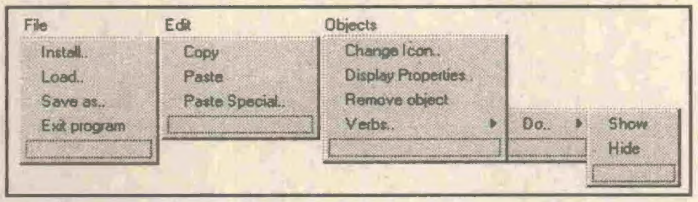
procedure TFoleDem.ChangeIcon1Click(Sender: TObject); กิจกรรมสำหรับเปลี่ยนรูป Icon ให้กับ OLE Object

procedure TFoleDem.DisplayProperties1Click(Sender: TObject); กิจกรรมสำหรับแสดงค่า property สำหรับ OLE Object ในปัจจุบัน

procedure TFoleDem.Removeobject1Click(Sender: TObject); กิจกรรมสำหรับยกเลิก OLE Object ที่ใช้งานอยู่



รูปที่ 5
หน้าต่างฟอร์ม
FoleWnd



รูปที่ 6 เมนูการใช้โปรแกรม OLEDem

การแสดงเมนูของ OLE Object

เราสามารถกำหนดให้แอพพลิเคชันที่เรียกใช้ OLE Object แสดงเมนูของ OLE Object โดยการเพิ่ม TmainMenu Component เพื่อเชื่อมโยงเมนูของ OLE Object เข้าสู่ระบบโดยอัตโนมัติ ในกรณีที่แอพพลิเคชันมีเมนูอยู่แล้ว การกำหนดให้เมนูที่มี GroupIndex เท่ากับ 1, 3, 5 จะถูกยกเลิกทันที และแทนที่ด้วยเมนูของ OLE Object

สำหรับการแสดงผลของ Tool Bar หรือ Binder ที่แสดงผลอยู่ ณ ตำแหน่งขอบของหน้าต่าง จะต้องเพิ่ม Tpanel Component โดยกำหนดให้ align ไปที่ขอบของหน้าต่างที่ต้องการ เช่น alTop, alBottom, alLeft หรือ alRight

หน้าต่าง MDI Child : uOLEWnd

uOLEWnd.pas ดังรูปที่ 5 แสดงอยู่ในลิสติงที่ 2 เป็นหน้าต่างที่บรรจุ ToleContainer เพื่อใช้ติดต่อกับ OLE Object ต่าง ๆ ที่มีอยู่ในระบบ การ activate ออบเจ็กต์เพื่อป้องกันข้อผิดพลาดและเข้าใจโดยง่าย ผมจึงใช้กิจกรรม InsertObjectDialog สำหรับ Embedding และ Linking ไปยังออบเจ็กต์ที่ต้องการ โดยได้ทำการแบ่งคำสั่งไว้เป็น 2 ชุด ชุดแรกจะเรียกผ่านเมนูจากหน้าต่างหลัก (uOLEDem.pas) และชุดที่สองเป็น Popup Menu อยู่ภายในฟอร์ม FoleWnd เอง มีกิจกรรมที่สำคัญคือ

procedure TFoleWnd.Installobject1Click(Sender: TObject); กิจกรรมสำหรับเลือก Embedding หรือ Linking Object เข้าสู่ระบบ

วิธีการใช้โปรแกรม OLEDem

การใช้โปรแกรม OLEDem จะใช้การควบคุมผ่านระบบเมนู ดังรูปที่ 6 มีรายละเอียดดังต่อไปนี้ครับ

เมนู File

- Install.. เลือกเมนูนี้เมื่อต้องการ Embedding หรือ Linking ไปยัง OLE Object รายการใหม่ (สามารถเปิดได้สูงสุด 10 รายการ)

- Load.. อ่านข้อมูลชนิด OLE Object File (.OLE) จากสื่อบันทึกข้อมูลที่ได้เคยบันทึกไว้

- Save as.. บันทึกข้อมูลทั้งชนิด Object File (.OLE) หรือตามชนิดเอกสาร (.DOC, .VSD, Etc.) โดยการเลือกชนิดที่หน้าต่าง Save Dialog

- Exit Program ออกจากโปรแกรม

เมนู Edit

(ปกติเมนูนี้จะถูกแทนที่ด้วยชุดเมนู Edit ของ Object ที่ติดต่อด้วย)

- Copy (to clipboard) สำหรับ copy ข้อมูลจาก Object ไปยัง clipboard ของระบบ (Windows) - Paste

- Paste (to clipboard) สำหรับ paste ข้อมูลจากแหล่งอื่นที่บันทึกใน clipboard ให้กับ OLE Object (จะต้องเป็นข้อมูลชนิดเดียวกัน)

- Past special.. สำหรับ paste ข้อมูลจาก clipboard โดยสามารถเลือกชนิดของ source data ได้ว่าเป็นข้อมูลชนิดไหน

เมนู Objects

(เมนูที่ควบคุมการใช้งาน Objects ทุกตัวที่สร้างขึ้น)

- Change Icon สำหรับเปลี่ยนรูป Icon ของ OLE Object ในกรณีที่ไม่ต้องการให้แสดงตัวบน Form

- Display Properties สำหรับแสดงค่า Properties หรือข้อมูลต่าง ๆ ที่บันทึกไว้สำหรับ Object ปัจจุบันออกหน้าจอ

- Remove Objects สำหรับยกเลิกการใช้งาน Object ปัจจุบันที่ activated อยู่

- Verbs สร้างไว้เป็นตัวอย่างการเรียกใช้กิจกรรม Doverb มีรูปแบบการแสดงผลอยู่ 2 แบบคือ

- Show ใช้สำหรับ activated หน้าต่าง OLE Object

- Hide ใช้สำหรับ deactivated หน้าต่าง OLE Object

ส่งท้าย

สำหรับหัวข้อเรื่อง OLE Automation ที่เกี่ยวข้องกันคงต้องเก็บไว้คุยกันต่อภาคสองฉบับหน้า ในฉบับนี้ผมอยากขอให้นักอ่านท่านผู้สนใจลองเล่นโปรแกรม OLEDEM ดูสักครั้ง ท่านจะเห็นความมหัศจรรย์ของเทคโนโลยี OLE

ในอนาคตอันใกล้นี้ผมเชื่อว่าคงมีโอกาสได้เห็น Operating System ในรูป OLE Base เต็มรูปแบบอย่างแน่นอน ฉบับนี้ผมคงต้องกล่าวคำอำลาเพียงเท่านี้ สวัสดีครับ

ลิสติงที่ 2 โปรแกรม uOLEWnd.pas

```
00001: // *************************
00002: // uOLEWnd.PAS
00003: //   Release date : 11/11/41
00004: // *************************
00005: unit uOLEWnd;
00006:
00007: interface
00008:
00009: uses
00010:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00011:   OleCtnrs, Menus;
00012:
00013: const
00014:
00015:   MAX_OLE_CLIENTS = 10; // max. possible open form
00016:
00017: type
00018:   TFoleWnd = class(TForm)
00019:     OleContainer1: TOleContainer;
00020:     PopupMenu1: TPopupMenu;
00021:     Installobject1: TMenuItem;
00022:     procedure OleContainer1Activate(Sender: TObject);
00023:     procedure Installobject1Click(Sender: TObject);
00024:   private
00025:     { Private declarations }
00026:   public
00027:     { Public declarations }
00028:   end;
00029:
00030:   // define array of form
00031:   TFoleWndS = array[ 1..MAX_OLE_CLIENTS ] of TFoleWnd;
00032:
00033: var
00034:   FoleWnd: TFoleWndS;
00035:
00036: implementation
00037:
00038: {$R *.DFM}
00039:
00040: //---------------------------------------
00041:
00042: procedure TFoleWnd.OleContainer1Activate(Sender: TObject);
00043: begin
00044:   caption:=oleContainer1.SourceDoc;
00045: end;
00046:
00047: procedure TFoleWnd.Installobject1Click(Sender: TObject);
00048: begin
00049:   caption:='Loading..';
00050:   Application.ProcessMessages;
00051:   with OleContainer1 do
00052:   begin
00053:     if ( InsertObjectDialog ) then Doverb( ovShow );
00054:   end;
00055: end;
00056:
00057: end.
```